ORBexpress RT Product Review Package

Presented to the DII COE Real-time TWG
24 August 2000

Prepared by
Steve Grimaldi (grimas@ois.com)
Objective Interface Systems, Inc.

Product Identification

- **This design review package describes the ORB*express* RT ORB libraries and CORBAservices Name Service and Event Services to be supplied to the DII COE as extended toolkits.**

- **Language bindings supported: C++, Ada95**
- **Target DII COE platforms: LynxOS 3.1.0a, Solaris 8**

- **ORB*express* RT is sponsored by the United States Air Force on behalf of the RT TWG and the following programs with plans for its use:**

  - Joint Tactical Radio System (JTRS)
  - Advanced Field Artillery Tactical Data System (AFATDS)

Question 1. Architecture

How does the component work as an application? What is the design of the component? Is there a server piece? A client piece? Does it operate standalone? Does it have connectivity requirements, etc?

Question 1. Architecture

- ORB library – linked to the application by the developer in order to create an executing ORB*express* RT application. This library is used to create both CORBA Clients and CORBA Servers. The ORB*express* RT library uses the operating system supplied TCP/IP library to communicate with other CORBA programs.

- IDL compiler – used by the developer to translate OMG standard IDL into programming language source code. ORB*express* RT uses separate IDL compilers for Ada and C++.

- Name Service – a separate program which executes at run-time, providing OMG standard Name service.

- Event Service - a separate program which executes at run-time, providing OMG standard Event service

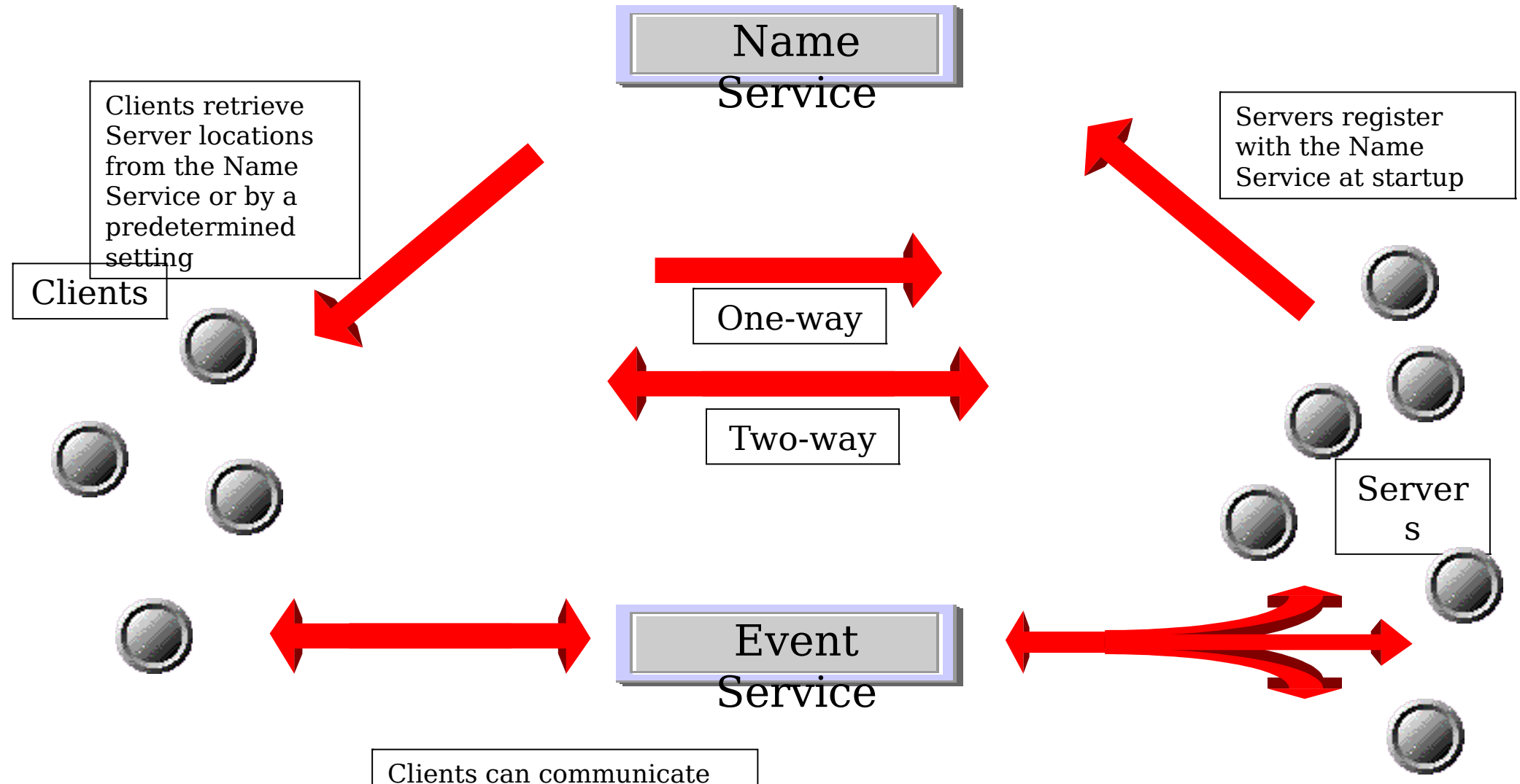
Question 1. Architecture

Separate ORB*express* RT products are supplied for each language, compiler and operating system combination.

For the DII-COE RT TWG:

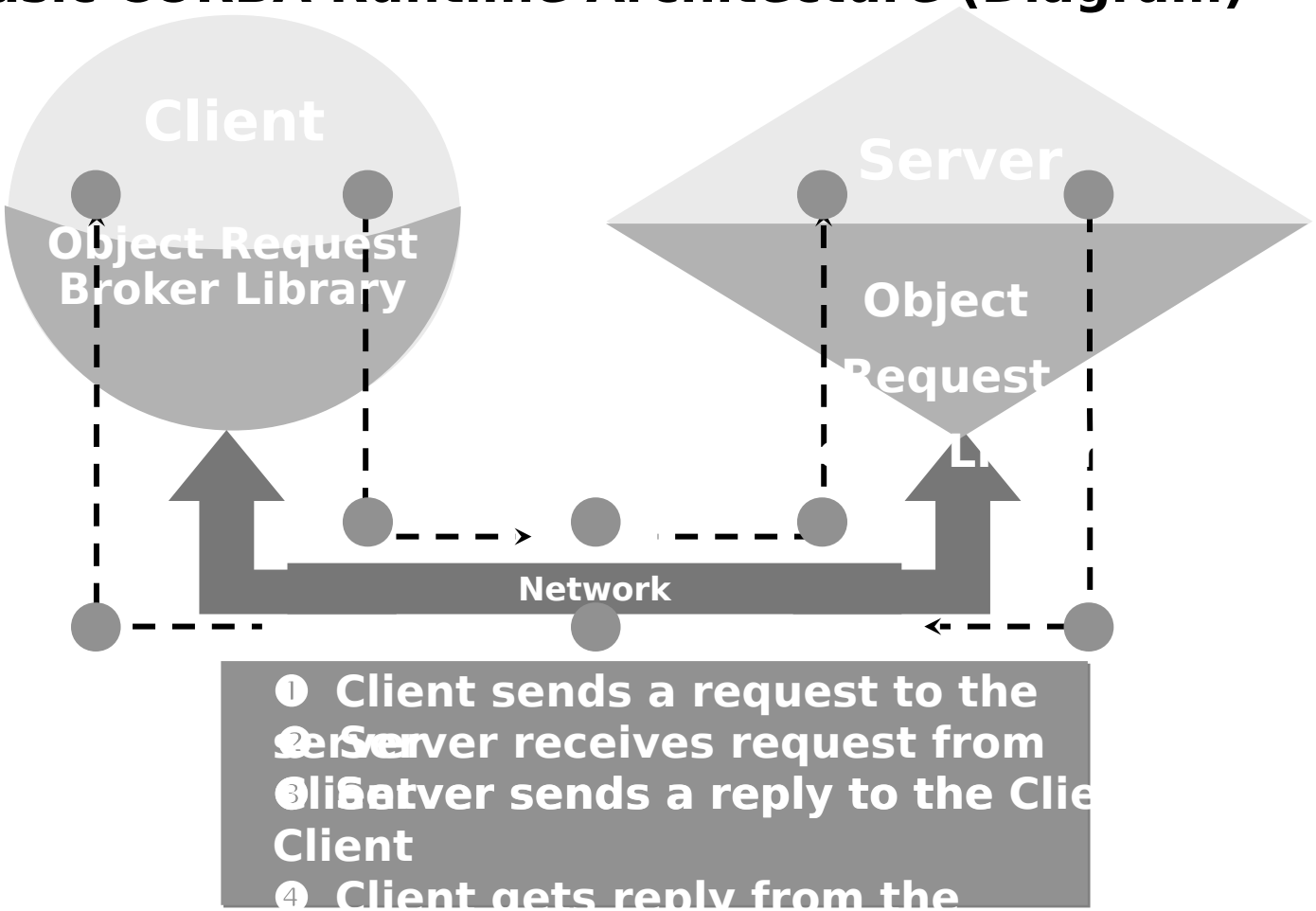
- ORB*express* RT Ada Solaris Gnat
- ORB*express* RT Ada Lynx Gnat
- ORB*express* RT C++ Solaris SunCC
- ORB*express* RT C++ Lynx GCC

CORBA Application


Name Service
Clients retrieve Server locations from the Name Service or by a predetermined setting
Servers register with the Name Service at startup
Clients
One-way
Two-way
Servers
Event Service
Clients can communicate directly or through the Event Service

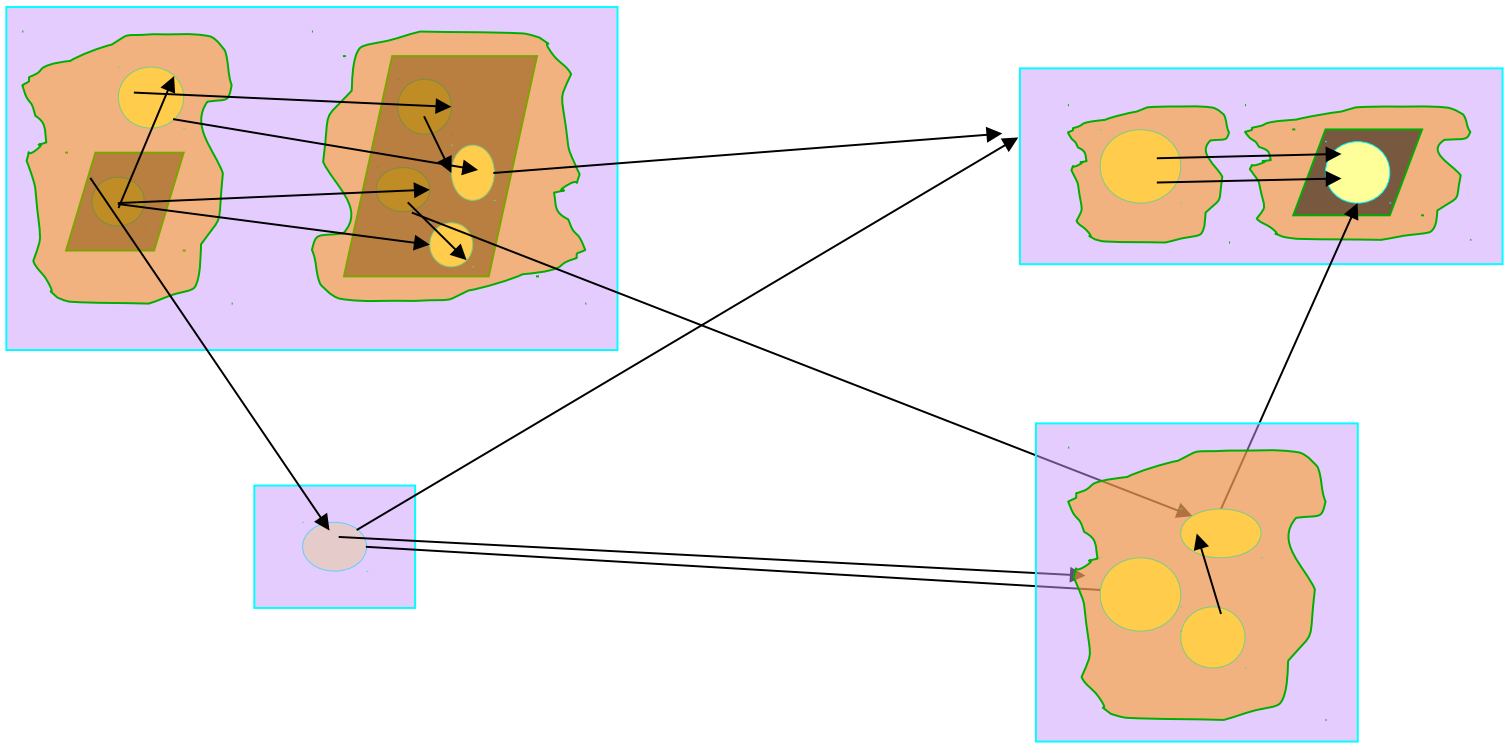
CORBA Runtime View Overview

Basic CORBA Runtime Architecture (Diagram)



Distributed System

- **Distributed Objects can be easily moved**
  - Process Local - same address space, same machine
  - Machine Local - different address space, same machine
  - Remote – different address space, on a different machine



ORBexpress Resource Requirements

Resource estimates with respect to memory, tape, disk, performance, data file sizes, etc. For each supported host, what are the minimum resources required? Maximum? Average/typical?

	Component	ORB*express* RT C++	ORB*express* RT Ada
Memory	How many kilobytes minimum? If applicable formula for size requirements considering application use	100k library size 300 bytes/object 16k bytes/conn.	1.3MB library size 500 bytes/object 64k bytes/conn.
Processors	Minimum processor configuration as well as recommended additional capacity; identify processor type/families	any PowerPC or Sparc processor	any PowerPC or Sparc processor
Systems Disk	Number of megabytes for storage of program executables and static data in target system	0 bytes	0 bytes
Disk	Number of megabytes for storage of dynamic application data in target system	0 bytes	0 bytes
Network Connectivity	TCP/IP – IIOP - application specific transports can change the inter-object overhead	GIOP Header 40 - 80 bytes	GIOP Header 40 - 80 bytes

ORBexpress Resource Requirements

Resource estimates with respect to memory, tape, disk, performance, data file sizes, etc. For each supported host, what are the minimum resources required? Maximum? Average/typical?

	Component	ORB*express* Name Service	ORB*express* Event Service
Memory	How many kilobytes minimum? If applicable formula for size requirements considering application use		
Processors	Minimum processor configuration as well as recommended additional capacity; identify processor type/families		
Systems Disk	Number of megabytes for storage of program executables and static data in target system	**To Be Supplied**	
Disk	Number of megabytes for storage of dynamic application data in target system		
Network Connectivity	TCP/IP – IIOP - application specific transports can change the inter-object overhead		

Question 3. Network Use

Discuss how the segment is distributed around the LAN/WAN, what servers it expects to access and the flow of how configuration and execution takes place. If client/server, how does the client find the server, etc.? Discuss bandwidth utilization issues and the impact of specific implementation (Broadcast, Push/Pull, etc...).

Question 3. Network Use

Event Service (TCP) - when used provides a Publish /Subscribe abstraction. This allows the Client to be freed from managing multiple Subscribers, as well as being blocked by them.

Cost - duplication of message traffic N times, where N is the number of Server Subscribers to a particular Client. Also, some minor overheads to establish both publish and subscribe relationships at initialization.

Name Service - Clients and Servers can find each other using the Names Service (directory) or by direct addressing using a fixed network address and port (TCP/IP). When the Name Service is used, the direct address is supplied to the Client ORB which then communicates directly to the server side ORB.

Cost – used typically at initialization, or possibly fail-over

Question 3. Network Use

GIOP - Additional bandwidth used by a CORBA application is governed by the General InterOrb Protocol (GIOP) message format.

- ORB*express* RT uses IIOP (GIOP over TCP/IP) exclusively for all inter-ORB communications.
- GIOP overhead is message/application specific (40 – 80 bytes per message)

ORB*express* RT provides the ability to replace TCP with other protocols that might be required by Real-Time applications (such as VME, or RACEWAY). This gives the developer the opportunity to achieve different Real-Time semantics than those provided by TCP, and still use an Open System standard.

- ORB*express* RT uses the interoperable message format (GIOP) between clients and servers even when the underlying protocol is something other than TCP/IP. This preserves the standard even over uncommon communications media.

Question 4. Security Architecture

Discuss as applicable (strategy for RWX permissions on files, auditing, etc.) Is the product intended for use in classified systems? Is it intended for use in multilevel security environments? Are there any aspects of the design, implementation, or behavior of the product that have adverse implications for its use in classified or multilevel security environments? Is the product classified, and, if so, what is its classification?

- ORB*express* RT files are installed by a System Administrator with associated permissions by that user ID
- ORB*express* RT has no specific security constraints or considerations.
  - -many customers work in classified environments
- -Developers must evaluate the security needs of their applications.
- ORB*express* RT does not depend on any files or directories with global write permission.
- By replacing IIOP with a customer defined transport, secure communications can be achieved (custom or vendor supplied).
- ORB*express* RT does not create any files.
- The application can direct the ORB*express* RT Name Service to store the contents of the name space. The developer controls the location of this file, as well as its security settings.

Question 5. Segment Format

Discuss how this software is packaged into segments. How the segment is actually formatted (i.e. Segment types, etc.). Is the product available in binary executable image, shared runtime library (dynamically linkable), statically linkable object library, and/or source code formats?

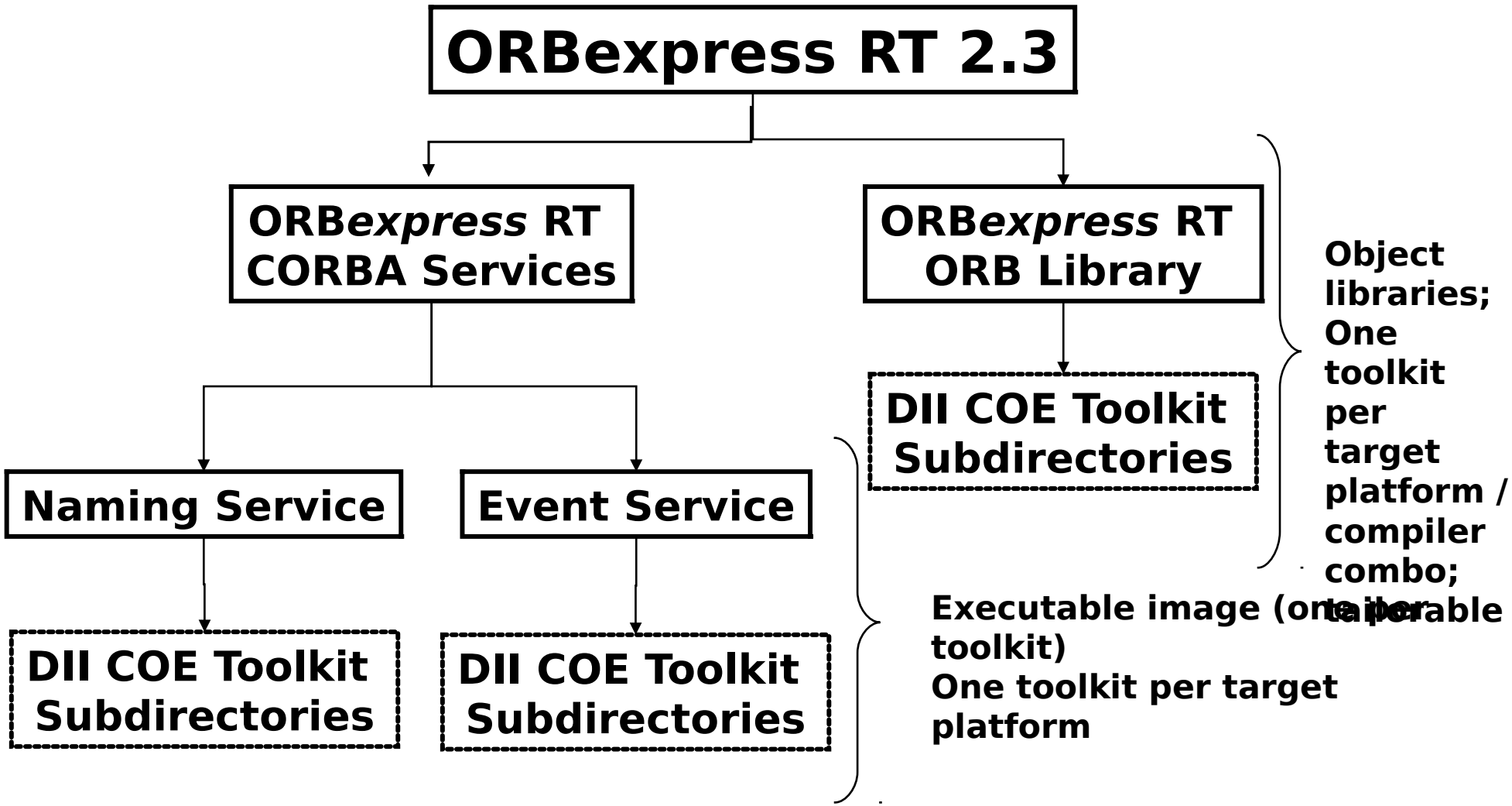
This effort has produced extended toolkits for ORB*express* RT statically linkable ORB libraries on Solaris and LynxOS.

Separate extended toolkits have been created for the executable Name Service and Event Service.

ORBexpress Extended Toolkit Format

Extended Toolkit Name	Ext. Toolkit Prefix	OS	Type of Segmentation	Segment Type	Version	COE Version
ORBexpress RT Ada	OERTAD	Solaris Lynx	Extended Toolkit	COTS	2.3.X	1.0.0.0
ORBexpress RT C++	OERTCX	Solaris Lynx	Extended Toolkit	COTS	2.3.X	1.0.0.0
ORBexpress RT Naming Service	OERTNS	Solaris Lynx	Extended Toolkit	COTS	1.1	1.0.0.0
ORBexpress RT Event Service	OERTES	Solaris Lynx	Extended Toolkit	COTS	1.0	1.0.0.0

ORB*express* RT Extended Toolkits


ORBexpress RT 2.3
ORBexpress RT CORBA Services
ORBexpress RT ORB Library
Naming Service
Event Service
DII COE Toolkit Subdirectories
DII COE Toolkit Subdirectories
DII COE Toolkit Subdirectories
Object libraries; One toolkit per target platform / compiler combo; interoperable
Executable image (one per toolkit)
One toolkit per target platform

Question 6. COE Compliance

COE Compliance: Target level of COE compliance, with strategy to achieve level 7/8

- Target level of COE compliance for ORB*express* RT is level 8

Question 7. Standards Compliance

List of standards followed by the segment (e.g. POSIX).
Identify and describe all proprietary interfaces

The ORB*express* RT product implements the OMG RT CORBA 1.0 standard as well as the OMG CORBA 2.3 specification

POSIX interfaces to the operating system are used

ORB*express* RT Standards Compliance

JTA Section	Description	Standard	Status
2.2.2.2.1.11.2	Distributed-Object Computing	OMG documentformal/98-12-01. Common Object Request Broker: Architecture and Specification, Version 2.3, June 1999.	Y
2.2.2.2.1.11.2	Distributed-Object Computing	OMG document formal/97-12-10. CORBAservices Naming Service Specification, March,1995.	Y
2.2.2.2.1.11.2	Distributed-Object Computing	OMG document formal/97-12-11. CORBAservices Event Service Specification, March 1995.	Y
2.2.2.2.1.11.2	Distributed-Object Computing	OMG document formal/97-12-17. CORBAservices Transaction Service Specification, November 1997.	N/A
2.2.2.2.1.11.2	Distributed-Object Computing	OMG document formal/97-12-21. CORBAservices Time Service Specification, July 1997.	N/A
2.2.2.2.1.11.2	Distributed-Object Computing	OMG document formal/97-12-23. CORBAservices Trading Object Service Specification, March, 1997.	N/A

Note: approval items noted in response to question 18 for all N/A items.

ORB*express* RT Additional Standards Compliance

JTA Section	Description	Emerging Standards Required by TWG	Status
2.2.3.5.2	Distributed-Object Computing	OMG TC document/99-02-12, Real-time CORBA Joint Revised Submission, March 1999	Y
2.2.3.5.2	Distributed-Object Computing	OMG document orbos/99-03-29, Errata for the Real-time CORBA Joint Revised Submission	Y

JTA Section	Description	Additional Emerging Standards	Status
2.2.3.5.2	Distributed-Object Computing	OMG document orbos/98-03-04/CORBAservices Interoperable Name Service	Y

ORB*express* RT Standards Compliance

Standards Governing POSIX Conformance

JTA Section	Description	Standard	Status
2.2.2.2.1.7	Operating System Services	ISO/IEC 9945-1:1996, Information Technology - Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) [C language], including real-time and Thread extensions	Y
2.2.2.2.1.7	Operating System Services	ISO/IEC 14519:1999, Information Technology - POSIX Ada Language Interfaces - Binding for System Application Program Interface (API) - Realtime Extensions	??

Note: approval items noted in response to question 18 for all N/A items.

Question 8. External Software Requirements

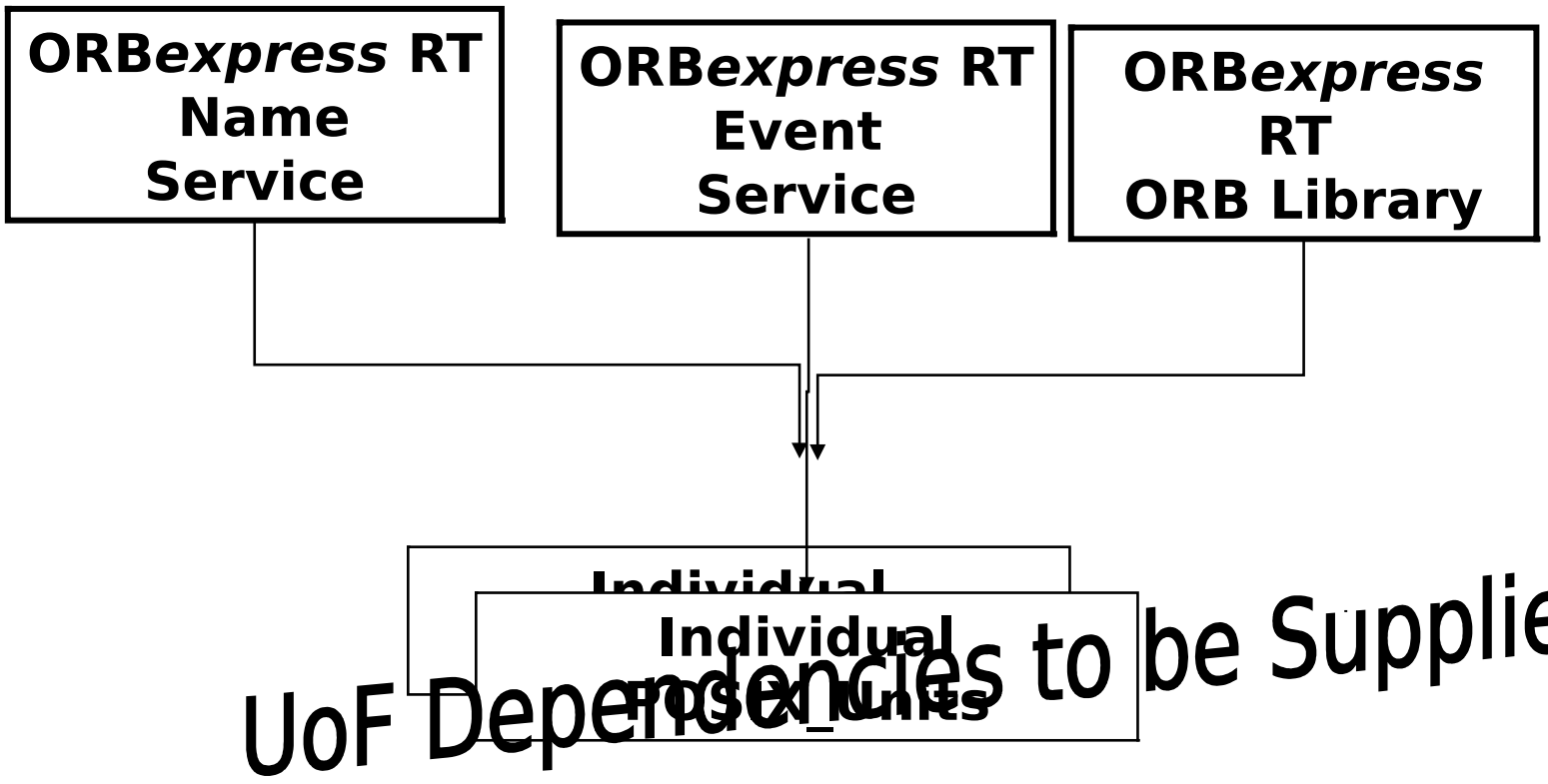
List of all COTS, or OS items required by the segment to make it work. This list should include commercial version numbers for the COTS products. This list should cover, which OS are covered by the product (i.e. Solaris 2.5.1, HP-UX 10.20, and NT 4.0, etc.).

- Lynx Operating System – version 3.1.0a (for PPC processor)

- Solaris Operating System – version 8 (for Sparc processor)

Question 9. Segment Dependencies

- **Discuss the dependencies and/or conflicts as documented in SegDescrip files. List and describe all installation, operational, runtime, and functional dependencies on other segments.**



Question 9. Segment Dependencies (cont.)

When configured to use the default IIOP protocol, ORBexpress products depend on the availability of the TCP/IP protocol and sockets as typically supplied by the operating system vendor.

ORB*express* RT introduces no dependencies into executable files produced for the target OS.

ORB*express* RT Naming Service and ORB*express* RT Event Service have no additional runtime dependencies.

Question 10. Functional Duplication

List of current segments that have functions/features the proposed effort may duplicate, and a rationale for why the duplication and strategy for resolving duplication.

ORB*express* RT products are being proposed for inclusion in the DII COE Infrastructure Services layer. They do not duplicate the functionality of segments currently in the COE, because there is no CORBA ORB in the COE at the present time. However, they **do** duplicate capabilities provided by several ORBS that have been segmented and are currently in use in major DII COE-based systems (TBMCS, for example).

Duplication of function is justified in proposing multiple ORBs as COE components. Each of the ORBs proposed covers needs that are not satisfied by other products of the DII COE. ORB*express* RT provides bindings for Ada and C++ programs.

ORB*express* RT interoperates with other ORBs that have already been segmented as DII COE segments while extending ORB behavior to the real-time domain. Moreover, the proliferation of ORBs will not affect either portability or interoperability of applications significantly, because key programmatic interfaces are controlled by the CORBA standard.

Question 11. Keyboard Mappings

List of all assumed keyboard mappings to be added to the system.

N/A - The ORB*express* RT product has no keyboards mappings of any kind

Question 12. Color Map Usage

Discuss any taking over of the palette, etc. If the products can use an extreme amount of colors, is it configurable?

N/A - The ORB*express* RT product has no color map usage of any kind.

Question 13. Menu/Icon Additions

Discuss any menus and or icons that are added to the system by the application.

N/A - The ORB*express* RT product has no menu or icon additions of any kind.

Question 14. Installation Process

Discussion of installation flow, especially with respect to duties for the segment's PostInstall and process when segment is replaced or updated in the field. This should really tie the segment format of #5 in with the architecture of #1. Is there an installation order, etc.? This will be impacted if/when DISA concurs that some COE components need to be made available to system developers (DoD contractors) in forms other than executable binary applications.

ORB*express* RT has no installation order requirements.

ORB*express* RT Ada, C++, Naming Service and Event Service have no installation order requirements.

The toolkits are simply ‘untarred’ to the desired location and a single environment variable is set.

Question 15. User Interface

GUI walkthrough (identify style guide issues)

N/A - The ORB*express* RT product has no GUI interface.

Question 16. Testing Requirements

Testing Requirements: Testing information (what testing has been done) Discussion of functional testing completed by sponsor. Discussion of certification testing completed, if applicable. To what standards and/or specifications has the product been tested? Approximately what percentage of test coverage (e.g., in terms of total executable instructions and execution paths) is provided by testing done to date? What certifications, if any, has the product earned? How many known errors exist in the version of the product being submitted to the DII COE? What is the nature of these known errors/problems?

A discussion of testing is to be supplied

Question 17. Risks

Risks associated with the segment (i.e. people, machine, and other resource limitations). Is the product certified for use in systems or situations where the product may, directly or indirectly, have an effect on the safety of persons or property? If so, list and describe all such certifications.

ORB*express* RT has not received any safety certifications.

Question 18. Approval Items

Approval Items:
List of any items which require Chief Engineer approval, as per the I&RTS. (Boot processes, background processes, and shared memory requirements, Root directive in Seg Info file, etc.)

Document Ref	Description of approval request
2-4	The LynxOS operating system, the baseline OS of the reference implementation of the DII COE RT Kernel, does not provide NIS/NIS+ capability.
6-8	Toolkits for ORBexpress RT have been provided for the Solaris, and Lynx environments. ORBexpress RT is also available for HPUX, and Windows NT. These environmets can easily be supported should the request be made.
JTA 3.0, Para. 2.2.2.1.11.2	ORBexpress RT does not provide the following services listed in the JTA. Time Service, Transaction Service, Trading Object Service are not required for RT CORBA, based on the analysis performed by the RT TWG.

List to be expanded to include documentation waivers, etc.

Note: where document identification is omitted, assume the items refer to the I&RTS Appendix B, specifically the draft version issued with interim guidance for DII COE compliance of real-time systems.

Question 19. Software Licensing Issues

Highlight COTS licensing required, if applicable.
Additionally, elaborate on the license scheme used by the product line.

ORB*express* is licensed by Named User Developer. Each product is issued a software key which is used during the installation process.

ORB*express* licenses only development seats. There are no run-time licenses or fees

.

ORB*express* RT Developer is anyone who uses:

- the IDL translator,
- the include files and packages during compilation
- the ORB library for linking

Executable programs created by the Developer are not licensed.

Therefore, applications delivered as completed executables need not be licensed. However, a system integrator who needs to link ORBexpress-based applications into executables must have an ORBexpress development license in order to link with the ORB libraries.

Question 19. Software Licensing Issues

Highlight COTS licensing required, if applicable.
Additionally, elaborate on the license scheme used by the product line.

- ORB*express* RT is licensed by developer by name for a specific ORB/language and platform (for example, for C++ on Solaris)
  - Development Cost: $12,500 (quantity 1)
    - Includes the first years maintenance
    - Product license is perpetual
  - Quantity discounts apply at unit 3 and above
  - Discounts for developers using more than one product
  - Product maintenance is annual
    - Applies to all compiler and OS updates
    - Any technical support or bug fixes
  - No run-time license or fee is applied

Question 20. Y2K Compliance

Is the component Y2K compliant? If it is not, then provide the plan for how and when the component will be compliant. If there are any question on Y2K compliance reference the Y2K web page.

ORB*express* RT is Year 2000 compliant

Objective Interface completed extensive tests prior to the Year 2000 as part of Y2K certification required by its customers

List of certifications needed? If so, in what form?

Question 21. Documentation

Based on the CM delivery checklist, which of these documents will be delivered, and which ones will need a waiver from the Chief Engineer. Also map the COTS documents to the appropriate CM requirements, so they will be understood at delivery time.

ORBexpress RT Documentation - Technical

"✓" or N/A*	Item	Comments
✓	Delivery Letter	Required
✓	Installation Procedures	Required
✓	Software Version Description	Required
See comments	System Administrator's Manual	COTS (Installation Guide)
See comments	User's Manual	COTS, includes Release Notes
N/A	Software Product Specifications	
N/A	Database Design Document	
See comments	Software Test Plan	Access available at vendors facility
See comments	Software Test Description	Access available at vendors facility
See comments	Software Test Report	Vendor format
✓	API Reference Manual	COTS
N/A	Programmer's Manual	Not delivered separately
✓	Software Design Description	Not Available
N/A	Interface Design Document	Governed by CORBA spec.

ORBexpress RT Documentation - Other

"✓" or "N/A*	Item	Comments
N/A	Errata Sheet	First delivery
✓	Printout: segname and VERSION files	Required
✓	Two (2) copies of executable software media	Required
✓	Two (2) License Agreements	Required
??	Intellectual Property Rights Agreement	
✓	Approved Waivers	Required

Question 22. Foreign Export Status

Is the component exportable to foreign entities?

- The software used for development license control in ORB*express* RT is classifiable under under an export control (5D992 for software and 5A992 for hardware).
- ORB*express* RT product may be exported to any country other than those which the U.S. has designated as terrorist threats, or for which there is a trade embargo.
- The current list is Afghanistan, Cuba, Iran, Iraq, Libya, North Korea, Serbia, Sudan and Syria.
- Executables produced using ORB*express* RT are not restricted and are available for export under license exception TSU (for technology and software, unrestricted).

Question 23. TWG Review Status

Have the applicable working groups reviewed the component? If so which working groups and what were the recommendations?

This briefing is being presented initially to representatives of the DII COE RT and DC TWGs. Their recommendations will be added to the briefing prior to presentation to DISA.

Background on the Upcoming Questions

- **Question 24. Assessment of Predictability**
- **Question 25. Schedulability Assessment**
- **Question 26. Maximum Blocking Time Analysis**
- **Question 27. Mechanisms to Bound Priority Inversion**

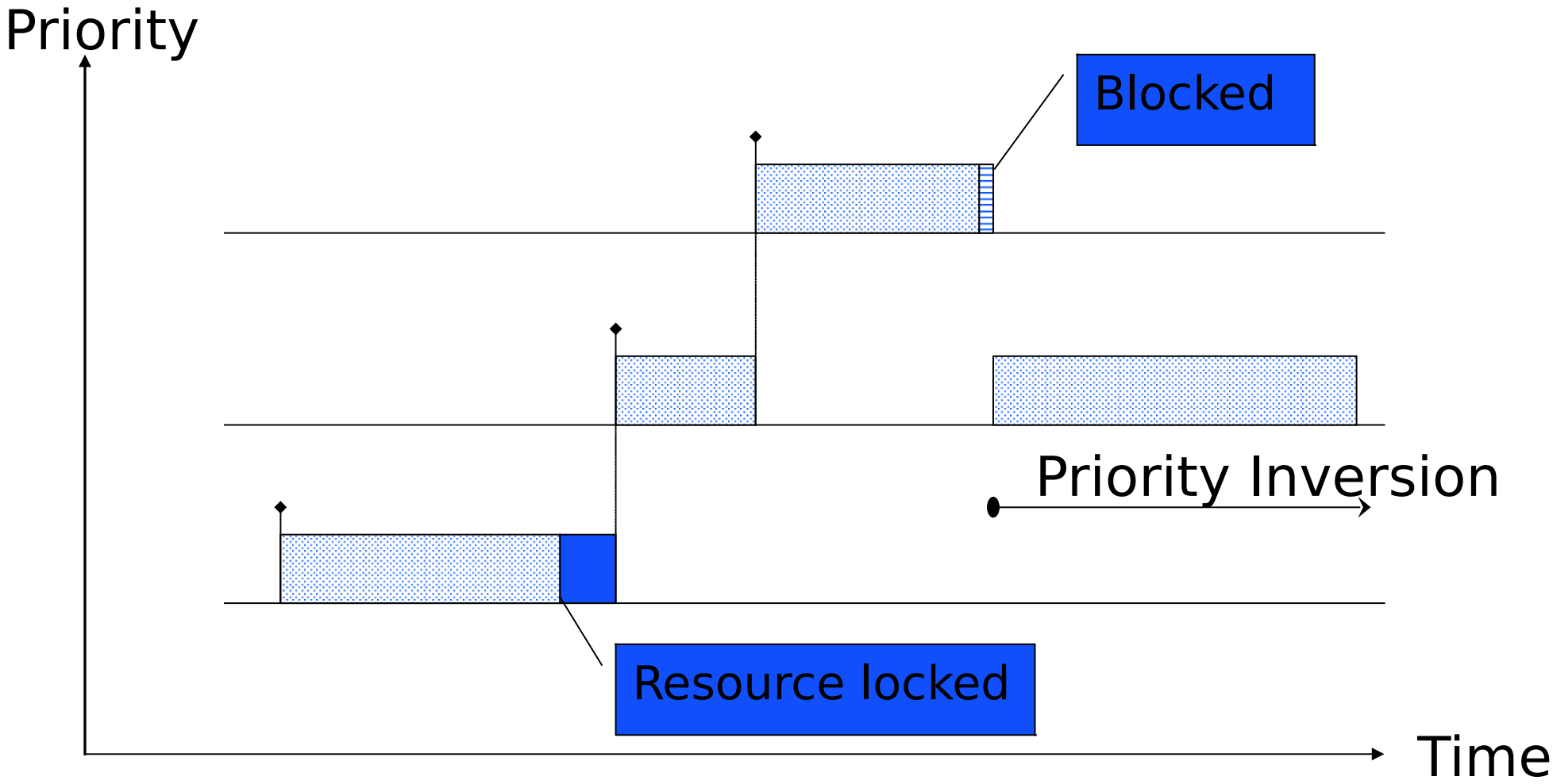
Real-Time Scheduling

- **Tasks**
  - Units of schedulable work
  - Assigned priorities
- **Real-Time Scheduling**
  - Preemptive – Lower priority tasks suspended when higher priority task enabled
  - Priority assignment related to the deadline
    - higher frequency implies higher priority
  - Fails if tasks miss deadlines

What Is Priority Inversion?

- **Whenever the highest priority task that is otherwise enabled is not executing**
  - Usual cause – contention for limited quantity resource with a lower priority task
  - For example, a lower priority task has locked a resource while in a critical section

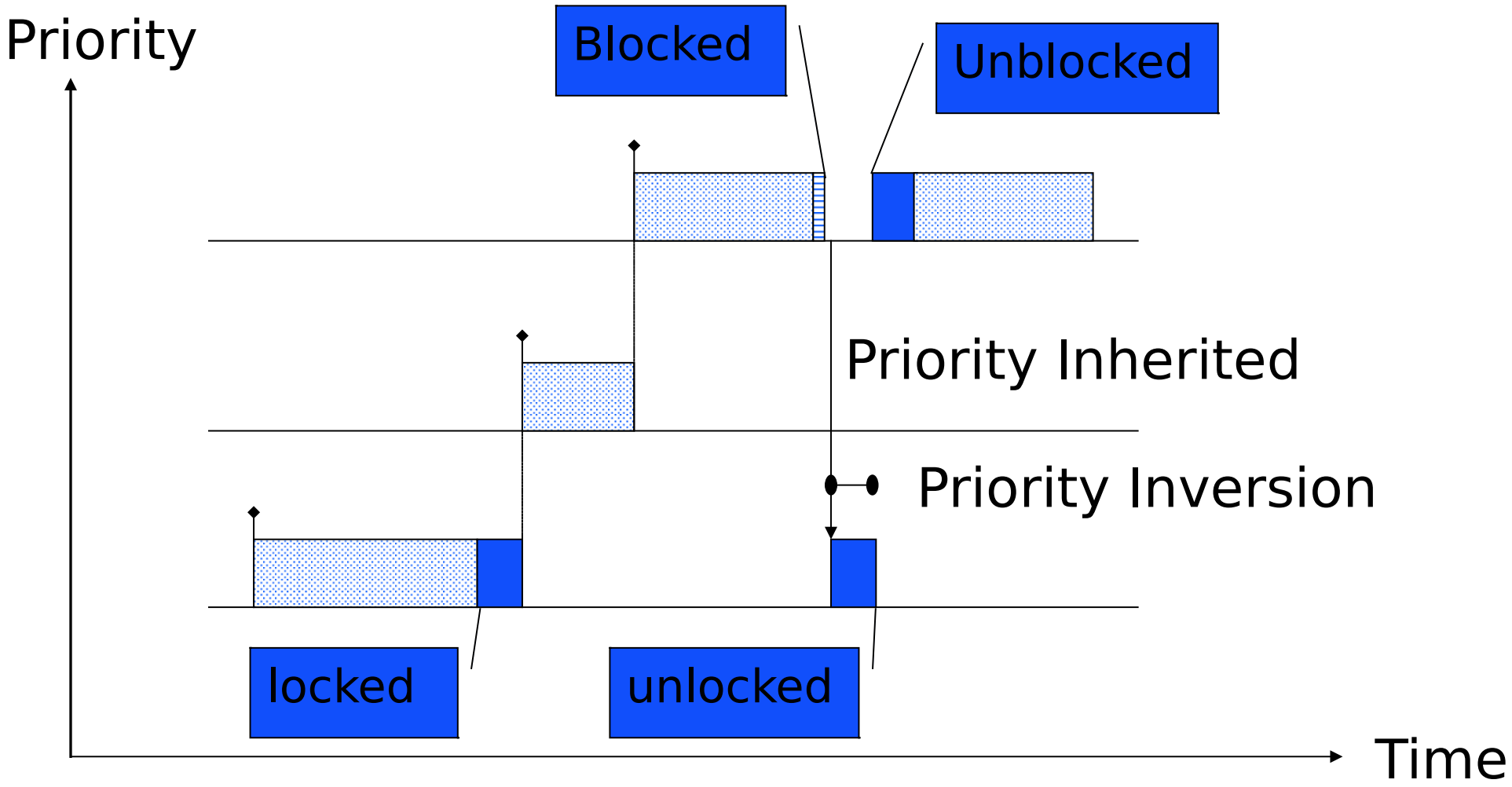
Example: Unbounded Priority Inversion



Distributed Priority Inversions

- **Occur when request processing on a remote node must contend for resources with other requests**
  - Other clients
  - Other threads
- **May require distributed priority inheritance to resolve**

Priority Inheritance



Bounded Priority Inversion

- **Priority inheritance helps to bound inversions**
- **But reduces schedulability of application**

$$R_i = C_i + B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

R_i – maximum response time of task i

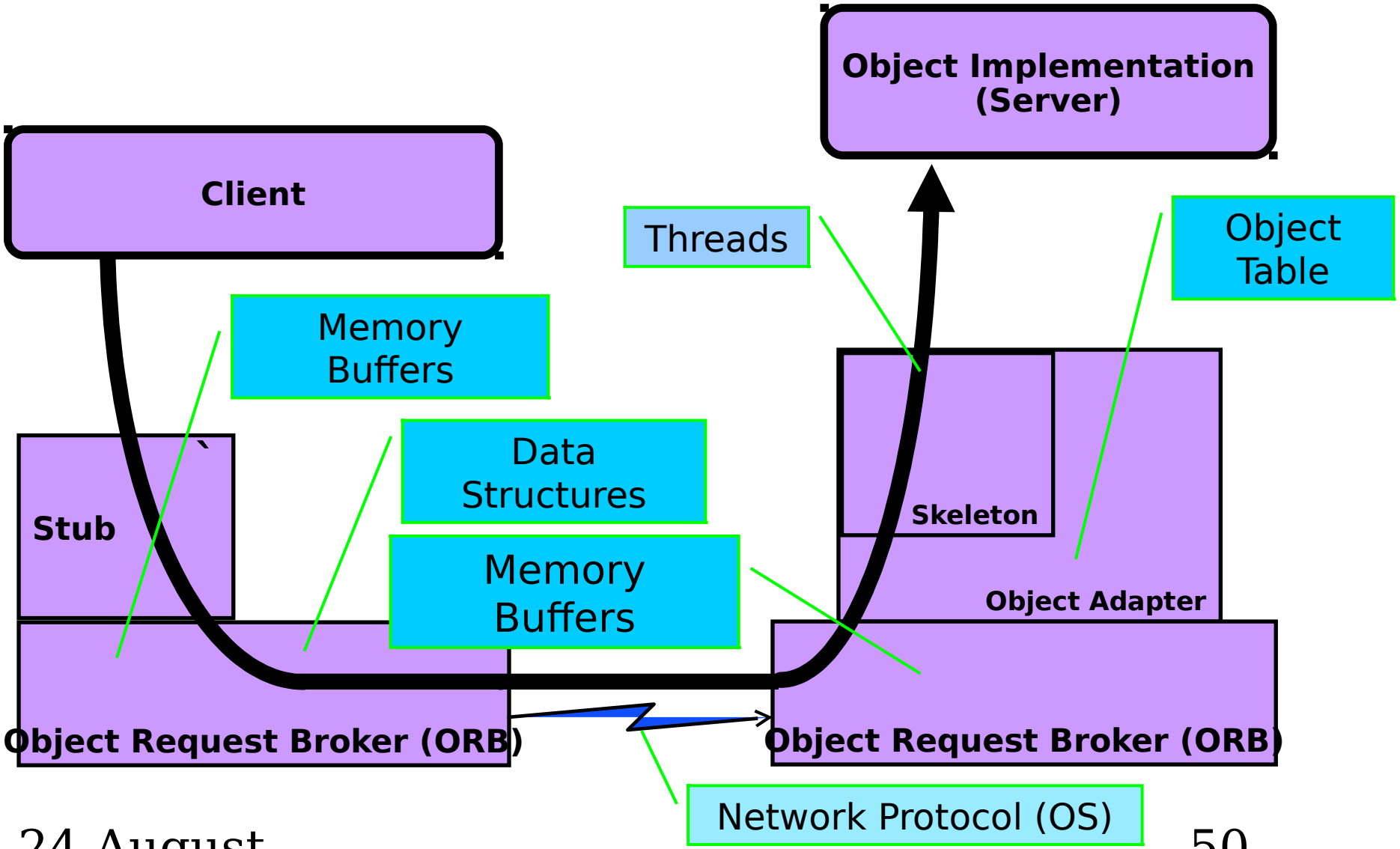
C_i – maximum computation time of task i

B_i – maximum blocking time of task i

T_i – period of task i

$hp(i)$ – higher priority tasks than task i

Sources of Priority Inversions in ORBs



Question 24. Assessment of Predictability

Describe how and under what circumstances the product achieves predictable time wise behavior. Identify features that do not offer predictable execution times.

Predictable behavior for any CORBA application is based on these factors:

- Application behavior
- ORB product behavior
- Operating System behavior
- Protocol stack behavior
- Network hardware behavior
- Plus all of the same on the server side

Question 24. Assessment of Predictability (cont.)

Predictable behavior for ORB*express* RT is accomplished by:

- **Inheriting priorities from the invoking application**
- **Using deterministic algorithms to implement internal data structures**
- **Using priority inheriting mutexes in ORB internals**
- **Applying deterministic algorithms to memory management**
- **Propagating priorities between clients and servers**

Question 25. Schedulability Assessment

Identify all schedulable entities (processes, threads) and shared resources. Identify scheduling parameters (e.g., priorities) and how they are established. Characterize triggering events and resource requirement. For each host platform supported (see question 1), list the minimum, maximum and typical resource requirements for each schedulable entity. Include the minimum and maximum number of each such entity that may exist simultaneously, or provide an algorithm for computing these numbers (e.g., 'one per mobile target track').

Scheduling is operating system specific

ORB*express* RT uses threads internally. their scheduling is controlled by the OS

Question 25. Schedulability Assessment (cont.)

Primary points of real-time scheduling is by:

- **Pre-emption according to priority and OS capability**
  - Solaris and Lynx provide different capabilities
- **At any mutex lock.**

Priorities are established by :

- **The calling application (client)**
- **Developer established priority bands (RT CORBA)**
- **The incoming request (server)**
- **Any lock conflict between resources**

Mutex are locked on (data structure) :

- Object table
- Connection table
- Priority map
- Memory data structure
- Thread list

Question 26. Maximum Blocking Time Analysis

Identify code regions in which preemption is prohibited. Identify upper bound on maximum non-preemptable time periods. For each non-preemptable critical region, for each supported host, provide maximum and typical cpu time durations during which preemption cannot occur, AND provide the number of times per second this critical region will be or may be executed. What is the average frequency of execution? Minimum interval between executions? Statistical distribution characterizing frequency of execution and/or execution time?

Preemption is prohibited during atomic mutex lock/release.

This can be computed by using:

- the number of instructions (2)
- the clock speed of the processor
- the number of clock cycles per instruction (processor specific)
- The number of wait states to memory (board specific)

Typical CPU time is measured in nanoseconds.

Question 26. Maximum Blocking Time Analysis (cont.)

- **ORB*express* RT data structures are selected and implemented based on determinism.**
  - Each ORB*express* RT critical section is less than 1000 cpu instructions.
  - The exact number varies
    - Based on compiler versions (both vendor, release specific).
    - Compilation switches (both vendor, release specific).
    - For example – versions of GCC generate different code, and have bugs
- **The number of times per second is application dependent**
  - based on the number of objects and active threads processing requests.
- **The primary factor for the frequency of non-preemption is transport efficiency (IO).**
  - Priority inversions introduced by the operating system are outside application control.
  - Previous versions of Solaris had priority inversions in the protocol stack.

Question 27. Mechanisms to Bound Priority Inversion

Identify mechanisms to prevent deadlock and to bound other scheduling anomalies such as priority inversion.

- Priority inversions are avoided by inheriting priority from the application, priority mutexes, and priority propagation.

- Deadlock prevention is accomplished through the use of priority inheritance, and ceiling. This depends on the priority bands set by the application. Each band has a lower and upper (ceiling) setting. So any ceiling setting is determined by application defined values.

Question 28. Real-time Performance Characterization

Characterize the real-time performance of this product: Timings, capacities, jitter, jitter tolerance?, etc.

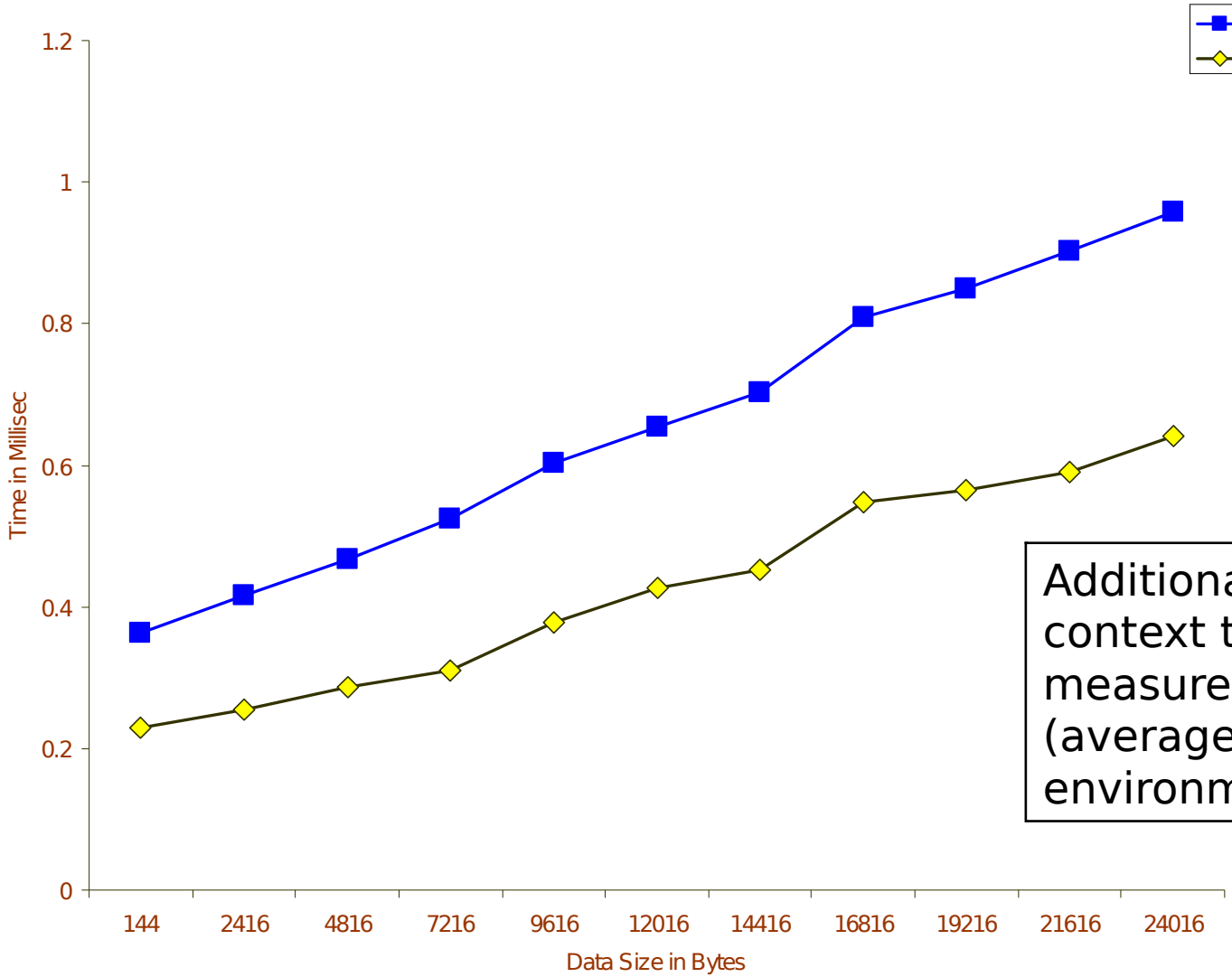
test type	min	ave	max	stddev
octet_type	57.46	194.20	45,057.87	718.00
long_type	55.27	189.80	79,233.37	868.00
double_type	91.49	227.80	13,949.37	225.00
any_type	93.32	230.40	25,233.51	415.00
unalignedFixStruct	93.11	244.60	15,258.94	293.00
octetArray1	86.46	245.60	43,954.06	518.00
	56.72	172.80	71,079.32	911.00

Additional benchmark context to be supplied: data sizes, for example

Solaris, one system, two-ways [TCP-Loopback] (time in microseconds)

Source: OE_Bench

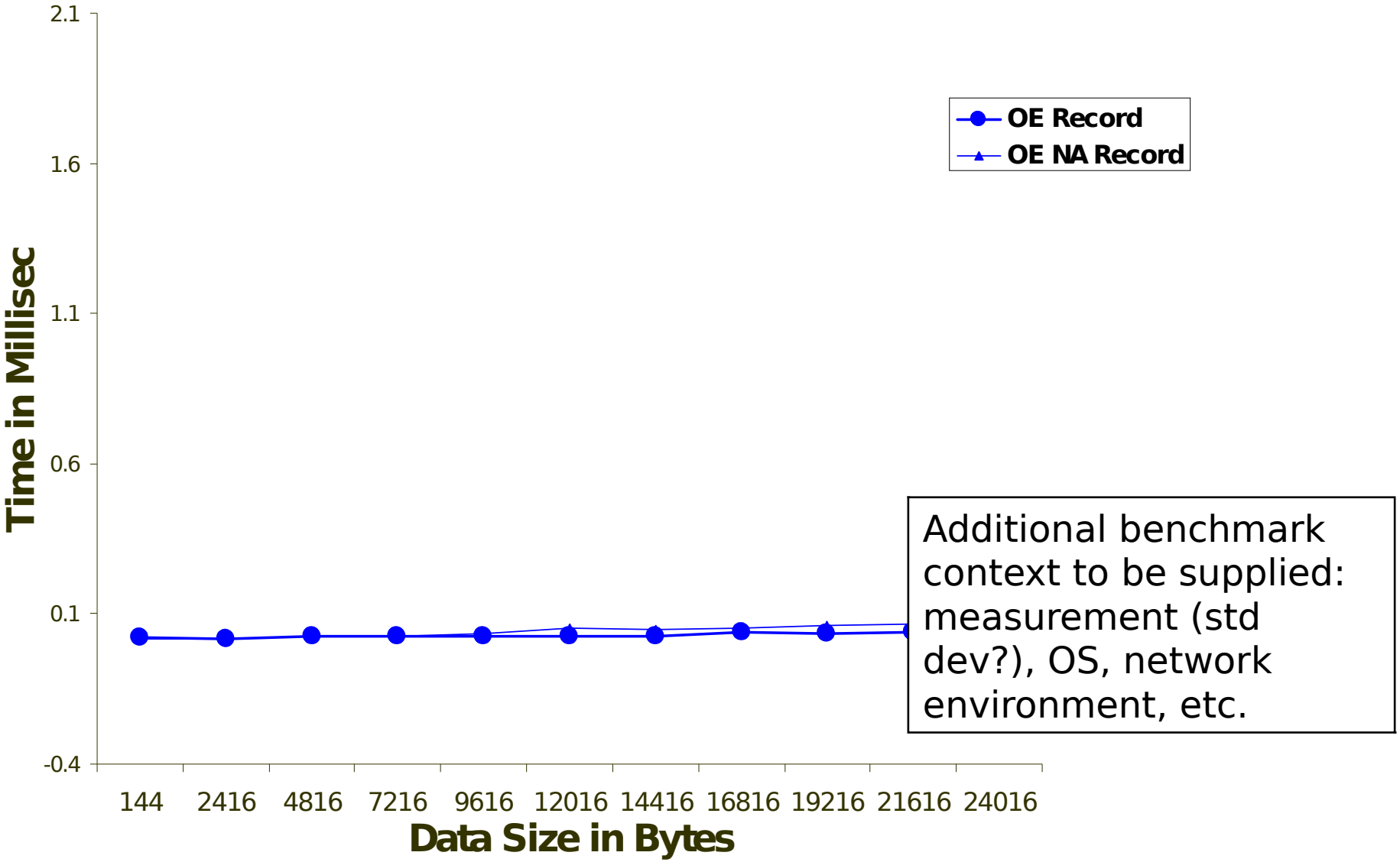
Performance



Additional benchmark context to be supplied: measurement (average?), OS, network environment, etc.

Source: DII-COE Study

Predictability



Additional benchmark context to be supplied: measurement (std dev?), OS, network environment, etc.

Source: DII-COE Study

Question 29. Multiprocessor Environment

If applicable, how will the segment work in a multiprocessor environment?

ORB*express* RT has been designed to take advantage of mulitprocessor environments, and has been tested extensively:

- Solaris, NT, Mercury, Concurrent, VxWorks
- Results show that the operating systems ability to schedule thread effectively and efficiently is the primary factor.
- Some protocol stacks are not multi-threaded

Question 30. Development and Integration Environments

Describe the preferred development environment (and options, if any). What assumptions does this product impose on the development environment (required or recommended tools), specific compilation systems, profilers, etc? What development tools does this package provide? What is development system cost, including cost of developer seat and run-time licenses?

Question 30. Development and Integration Environments

- The IDL compilers for ORB*express* RT Ada and CXX require a Java runtime for use during development only.

- Both Ada95 and C++ exhibit compiler dependencies, so the application using the ORB must be compiled using the same compiler used to build the ORB libraries. The extended toolkits to be delivered have the following compiler dependencies:
  - Ada 95 compiler - Gnat 3.13 (for Solaris and Lynx)
  - C++ compiler – SunCC 5.0 (for Solaris)
  - GCC compiler – GNU 2.7 (for Lynx)
- The GNAT and GNU compilers are freeware and have no licensing cost. Licensing costs for the SunCC compiler are available from Sun Microsystems.

- ORB*express* RT is licensed by developer
  - Development Cost: $12,500 (quantity 1)
    - Includes the first years maintenance

Question 31. Run-time Environment

What are the run-time requirements on operating system and kernel services (with respect to selectable services)? Are there mechanisms for configuring the product to improve responsiveness and predictability? For example, locking code or data in memory, adjusting scheduling parameters. Over what range can scheduling parameters be set, e.g., can "real-time" priorities be used by the application?

- ORB*express* RT can be configured to use Shared_Memory transport
  - –Significant performance improvement on same host transfers
  - –TCP irregularities are removed
    - Priority inversions, and predictability issues

- ORB*express* RT 'behaves' within the operating system settings
  - –Some OS' restrict real-time priorities to privileged users
  - –If the OS allows pages to be locked in memory, the ORB*express* RT behaves

Question 32. Interoperability

Can this product operate on other DII COE platforms (non-real-time or real-time) as well? Which ones? Characterize differences. Will this product interoperate with similar DII COE segments? What constraints are imposed? e.g., compatible IIOP for real-time and non-real-time CORBA products.

- ORB*express* RT has been independently proven to interoperate with other ORBS (such as ORBIX and TAO) at the message level (IIOP)

- Interoperability with GIOP compliant ORBs was addressed by the RT CORBA specification

- Developers must be aware of behavior issues when creating applications that interact with non-real-time ORBS. Real-time behavior cannot be expected when this kind of interface is involved.

ORB*express* RT Platform Coverage (General)

- **Host (and self targeted)**
  - Linux
  - Sun SPARC-Solaris
  - PC x86 Win95/98/NT
  - Dec Alpha Tru64 [self]
  - SGI MIPS-IRIX [self]
  - HP HPUX [self]
- **Target Processors**
  - Sparc
  - x86
  - Power PC
  - 68K
  - ARM
- **Compilers**
  - GNU g++, EGCS C++
  - Microsoft Visual C++
  - Sun WorkShop C++
  - DEC C++
  - Diab
  - Green Hills MULTI
  - SGI C++
  - Apex Ada
  - GNAT
- **Target OSs**
  - VxWorks
  - LynxOS
  - OSE
  - QNX Neutrino
  - Concurrent
  - Mercury